Visual Tracking Using HOG and SVM

1Deepthi V.K.P, Mr. Mohammed Anvar P.K 2
1PG Scholar, 2Assistant Professor, Al-Ameen Engineering College, Shoranur

Abstract
There exist a large number of visual tracking methods with revealing success. But the challenging problem in visual tracking is to handle the appearance changes of target object because of its adaptive ability. Difficulties in tracking objects can arise due to non-rigid motion, rapid movement, large variation of pose and scale, occlusion and drifts etc. One of the main reason for such failures is that, the unsuccessful image representation schemes of many algorithms. In this paper, we present a visual tracking method using HOG and SVM along with multiple kernel algorithm. The proposed work focuses on the implementation of the HOG features (Histogram of Oriented Gradients) to distinguish the target and the background with HOG visualization and facilitate classification using Support Vector Machine (SVM). In order to make our method more effective we are using a boosting technique to select good SVMs. We also include an update scheme to account for object appearance variance. Our tracker is able to handle occlusion and performs against existing visual tracking algorithms in handling various conditions. Qualitative and quantitative evaluations over various challenging sequences shows the competitive performance of our tracking algorithm.

Index Terms: Visual tracking, HOG, multiple kernel algorithm, SVM.

I. INTRODUCTION
In the recent decade, visual tracking has attracted so much attention because of its wide variety of applications. Visual tracking is used to determine the states of object (e.g: position, velocity, scale, and other related data) from images. However, the variable appearance of a target object due to factors such as pose, lighting, and shape deformation makes visual tracking challenging and thus other factors such as occlusion, non-rigid motion, rapid movement and background clutters further complicate this task. There are many existing methods of object tracking to address these problems using template matching, state estimation as well as foreground and background classification.

Template matching uses appearance model based on, intensity, statistical feature distribution, and low-dimensional subspace representations. Existing trackers based on template matching have demonstrated success in object tracking but they are less effective in handling large appearance change and tracker based on state estimation accumulates drift error during update and always fails in complex scenes. In the categorization based approaches, discriminating the target object from the background in successive frames. A super pixel tracking method created a discriminative appearance model based on super pixels and this method is effective in handling heavy occlusion and keep tracking under large appearance variations of target object. But the low tracking speed makes this tracker ineffective.

In this paper we present a visual tracking method based on the Multiple Kernel Boosting (MKB) algorithm along with Locality Affinity Constraints (LAC). Here we are using 3 feature descriptors RGB histogram, Histogram of Gradient (hog) and SIFT to describe an object. Also here we use linear kernel, polynomial kernel, RBF kernel and sigmoid kernel to map the input space to the kernel space. Here we give out all features to a group of different kernels instead of combining all features in the same kernel space. For each combination we train a single kernel SVM. Each single kernel SVM is considering as a weak classifier. The features classification performance on specific kernel-based classifiers can be evaluated thoroughly, which provides more ability for discrimination in the final decision function. We utilize boosting technique for efficiently selecting best combination of these svms. Affinity constraints related to the input data is computing and is adding to the final decision function.

The proposed SVM and HOG algorithm facilitates efficient update of classifiers to handle appearance change in object tracking. For updating, set of single kernel svms are retraining and using MKB reselecting some discriminative ones and recalculate LAC. We exploit affinity constraints to exploit the locality of data for learning good classifiers. Experimental results showing that our tracker is robust in complex scenes with good performance over existing methods and is faster comparing to the other tracking algorithms.

II. PROPOSED ALGORITHM
Our paper presents robust visual tracking in complex scenes effectively and efficiently using HOG and SVM along with multiple kernel algorithm. The HOG and SVM combination makes our system more efficient. Multiple kernel boosting trains a pool of classifiers, which are constructed by using different features. Boosting method works well as long as at least one kernel has enough discriminative ability, then. Boosting selects the strong classifiers in an active set and after each update, the number of selected classifiers may change because when tracking a object becomes difficult, MKB selects more classifiers. Three feature descriptors are using for describing object (RGB histograms, HOG, and SIFT), and four kernels (linear, polynomial, RBF, and sigmoid) are using for mapping input to kernel space. Experimental results demonstrate that different combinations of kernels and features with different weights providing effective visual tracking in different scenes.

A. Object Detection
Detect the object that we are interested to track using normal
detection method like warping of images. In the first frame, we are drawing a box enclosing the object and initializing the initial state of the target manually as $x^* = (c_1^*, c_2^*, s^*, 0^*)$ that records the position, size and rotation angle (superscript indicates the number of current frame). For increasing the training samples crop out a group of images for collecting positive samples i.e., $X^+ = \{x_i | 0 \leq l(x_i) - (l(x_i) - l(x_i)) < r_p \}_{i=1}^d$ and crop out a group of negative samples as $X^- = \{x_i | d \leq l(x_i) - (l(x_i) - l(x_i)) < r_t \}_{i=1}^d$.

Where $r_p$ and $r_t$ are constants. Setting $r_p > r_t$ for reducing overlap between positive and negative samples and $l(x)$ denotes the center of x. Note that we only use 1st frame to collect training samples. Now we have a set of training samples $\{x_i, y_i\}_{i=1}^D$, where $x_i$ denotes the $i^{th}$ sample and $y_i = \{\pm 1\}$ is the label corresponding to the sample.

**B. Image Representation and Feature Extraction**

SVM classifiers have been used in many classification and regression problems. The main difficulty in such tasks is to choose a suitable data presentation. In SVM based tracking methods, the data can be represented using kernels. For mapping the input space to the kernel space, we are using four kernels, linear kernel, polynomial kernel, RBF kernel and sigmoid kernel.

For any object there exist many features, interesting points on the object, which can be extracted for providing the feature description of the object. This description can then be used when attempting to track the object in an image containing many other objects. Here we are extracting features from the collected samples using RGB, HOG and SIFT feature descriptors. The RGB histogram is a mixture of three histograms based on the RGB color space’s R, G, and B channels. This RGB histogram having no invariance related properties. HOG descriptors counts occurrences of gradient orientation in the image. Scale-invariant feature transform (or SIFT) is an algorithm using for detecting and describing local features in an images. The extracted features are then sending to kernel. Then we get many combinations of features and kernels. For each combination, we train a single kernel SVM classifier. Thereby we obtain a pool of SVMs.

**C. Support Vector Machines**

In SVM-based methods, the data can be represented using kernel function $K(x, x_i)$, where $K(\cdot, \cdot)$ indicates a given positive definite function related to a reproducing kernel Hilbert space. Nevertheless, it is tough for a single SVM classifier to pick a good kernel for the given training dataset in some cases. To solve this problem, here we are using Multiple kernel algorithm to improve classification performance and to increase the interpretability of the decision function. Multiple kernel framework aims to find an optimal combination of multiple basis kernels and the related classifier. For explaining the concept of the Multiple kernel algorithm, we use a binary classification. Given a set training samples $\{x_i, y_i\}_{i=1}^D$, we have to classify unlabeled samples into a class by training a multikernel based classifier $F(x)$. Let $\{K_m\}_{m=1}^M$ be the kernel set computed for different feature. We can define the combination of multiple kernels as

$$K(x, x_i) = \sum_{m=1}^M \beta_m K_m(x, x_i)$$

**D. Multiple Kernel Boosting**

Multiple kernels methods cannot be directly applied to visual tracking due to several factors such as time-consuming optimization process, increased number of training samples and fixed weights. In this paper, we propose a boosting version of MKL (Multiple Kernel Learning) for feature selection namely, Multiple Kernel Boosting is used to decrease computational complexity and to increase accuracy of tracking applications. Existing boosting methods are not effective for tracking because they are designed to increase the accuracy of object categorization for a large number of instances rather than locality of object.

The decision function of our method can be written as

$$F(x) = \sum_{t=1}^L \beta_t h_t(x)$$

Where $L$ denotes the iteration times. Notation $m$ regarding the kernel function is replaced by 1 for denoting the index of iteration. Our MKB method does not entail solving a complex optimization, thereby making an efficient and effective algorithm for visual tracking.

As Fig. 1 shows, from the positive and negative samples of the training set, we are extracting $\{f_1, f_2, ..., f_n\}$ a set of features for sending into a set of $\{K_1, K_2, ..., K_M\}$ kernels, thereby obtaining $M \times N$ combinations. For each feature and each kernels on the training set, construct a pool of single kernel SVMs. Initialize the variable for sample’s weight value in order to take a decision as positive or negative support vector machine.

**Fig.1. Illustration of MKB process**
Calculate the weight variable for each frame or each set using the classifier’s classification error equation,
\[ C = \frac{\sum_{i=1}^{D} w(i). |h(x_i)|, U(-\gamma_i h(x_i))}{\sum_{i=1}^{D} w(i). |h(x_i)|} \]  
(4)

Here, \( U(x) = 1 \) when \( x \) greater than 0 and 0 otherwise, \( w(i) \) denotes weight of the training samples, and \( h(x_i) \) is the SVM’s real-valued classification output on the input \( x_i \). We select multiple features and kernels adaptively from the \( M \times N \) weak classifiers to form an optimal strong classifier with most discriminative ability.

### E. Object Tracking

MKB produces promising visual tracking results but in some situations it is not stable enough. Weight of the kernel depends on the relationship between input data and a particular function of kernel. So that the robustness of our MKB algorithm can be improved by distributing the training details into the final decision. For exploiting the underlying distribution of training data to track the target object, we devise a simple and effective method. We assume that individual SVM trained in the MKB algorithm have recorded the training data distribution on corresponding features and kernels. Here we factor the weight. Let \( \beta_i = \beta_i^* A_i(x) \) then we can write equation (3) as
\[ F(x) = \sum_{i=1}^{l} \beta_i^* A_i(x) h_1(x) \]  
(5)

Where \( \beta_i^* \) is same as in equation (3) and is computed by using MKB algorithm. \( A_i(x) \) is called Locality Affinity Constraint (LAC). Which is function of input \( x \) and representing similarity of trained SVM and input \( x \). This means that if an input sample has high bond with the distribution of a specific SVM classifier’s support vectors then that importance of that specific SVM is high. So we are assigning larger weight to that SVM classifier. We construct a model to describe the locality affinity constraints based on the probabilistic distribution as
\[ A_i(x) = 1 - \exp(-\sigma_i(x)) \]  
(6)

Where log odds ratio
\[ \sigma_i(x) = \frac{P(y=1|x)}{P(y=-1|x)} \]

For each \( h_1(x) \) (trained SVM), we calculate mean \( \mu^+ \) for positive vectors and \( \mu^- \) for negative vectors. Then we can compute \( P(y = 1|x) \) and \( P(y = -1|x) \) as follows
\[ P(y = 1|x) = \exp(-|x - \mu^+|) \]
\[ P(y = -1|x) = \exp(-|x - \mu^-|) \]
(7)

\( A_i(x) \in (0,1) \). \( A_i(x) \) can be seen as the probability of \( x \) belonging to the support vectors. We also makes simple and quick update of this affinity constraint by calculating only \( \mu^+ \) and \( \mu^- \). Our method is very suitable for visual tracking and comparing to other methods its computational complexity is low.

Important steps in our visual tracking method is shown in algorithm 1.

**Algorithm 1   Main Tracking Structure**

- **Input**: Training set, different feature, kernel functions for multiple kernels, and decision function whether classified or not.
- **Output**: Tracking result of each frame
  1. Read the initial frame.
  2. Detect the object that we are interested to track using normal detection method like warping of images.
  3. Obtain the initial stage and initialize the tracking parameters.
  4. From the detected stage extract the positive and negative samples
  5. Extract the different features like HOG, SIFT and RGB histogram from the initial stage.
  6. Train the frame for a single kernel SVM as per MKB process and generate a strongest classifier.
  7. For each new frame extract positive and negative samples.
  8. Calculate the weight function of each samples
  9. Use SVM output to generate the best classification result for each samples.
  10. Determine the tracking result by using SVM output. The tracking will be for positive sample and collect the remaining negative sample from the frame
  11. According to the negative sample choose the best SVM and update affinity constraints for each frame.
  12. Refresh the saved negative sample for each frame and this will help to track under occlusion.

We also include an update scheme in our tracking algorithm for handling object deviation. In each frame, consider the current...
state of tracked object $x^t$ as positive sample, and without overlapping with this sample extract 4 negative samples from four different directions (up, down, left, right). These samples consist mostly of the background region. So it is easy to discriminate foreground object from the background. For a few frames (e.g., 5 or 10 in this paper) we accumulate these samples in a queue and have enough number of new samples for retraining individual SVMs. Also we get updates of $A_1 (x)$ and $F (x)$. So that we can handle the most recent appearance variation of target.

III. RESULTS AND DISCUSSION

![Images of tracking results](image1.png)

Fig. 3. The results by our tracker when challenge encountered in tracking

We test our algorithm on different challenging sequences and the experimental results show that our tracker performs very well in complex scenes and is faster comparing to the other methods.

![Images of occlusion handling](image2.png)

Fig. 4. Illustration of effective occlusion handling

Fig. 3 demonstrate the tracking results of our tracker in complex scenes. Our method keep tracking the object even if there is large appearance variations. The bird of interest in fig.3 shows many poses such as turning around. From the figures we can see that our tracker is able to track the target in various poses with good performance. The HOG features are less sensitive to the illumination change. So that our method perform well even if there is light change. Also proposed method is designed to handle various motions such as fast movement and non-rigid motion as shown in fig.3.

Fig. 4 shows the effectiveness of handling occlusion. Where the pose of tom changes rapidly and finally occluded. Our tracker is able to track the target even if the target object is heavily occluded. Tracker keep tracking the head part of tom under severe occlusion as shown in frame #304. Since we use multiple kernels, we have a strong SVM classifier that facilitating a tracker to differentiate the target and background.

IV. CONCLUSION

In this paper we presents an efficient and effective robust visual tracking in complex scenes by creating an appearance model based on histogram of oriented gradients along with multiple kernel algorithm. The HOG-SVM combination is more effective and precise. It detects target objects even in difficult scenes. Using the extracted features and kernels of a frame, we make a group of SVM classifiers. Then selecting some of the strong classifiers from them for making the final decision using a multiple kernel boosting technique. This boosting technique is very effective in selecting good SVMs from a pool of classifiers. For further improving the robustness of our tracking, we are adding affinity constraints to the final decision thereby increasing the discriminative ability. An update scheme is also included to account for appearance variation of target object. Experimental results show that our tracking method performs very well comparing to the other methods in handling occlusion, rapid motion, harder background, scale changes and it is faster comparing to other methods. Most of the trackers keep tracking the object but tracking accuracy and speed of our tracker is better than the other methods.

REFERENCES


www.ijaert.org